Question 1 [20 Points]

In this problem, you are given the skeleton class `MatrixSearcher` (in MatrixSearcher.java). Your task is to implement the following method:

`public static boolean binaryFind(int[][] matrix, int target)`: Return `true` if `matrix` contains `target`, `false` otherwise. If `matrix` is null, simply return `false`. This method must implement the ***binary search algorithm***. You may assume that `matrix` is already sorted in ascending order prior to calling this method.

You may only use Java primitive functionalities, including loops, conditions, and arrays. You are not allowed to use built-in container classes (such as HashSet, HashMap, ArrayList, etc.). Use of recursion is encouraged but not required.

Hint: You can "flatten" out the matrix to a 1-dimensional array first before binary-searching the target value.

Test cases and expected output:

```
int[][] matrix = {{1, 4, 4, 7, 8, 10, 10, 10, 11, 11},
                  {12, 13, 14, 15, 15, 17, 17, 18, 18, 18},
                  {18, 22, 23, 23, 24, 24, 24, 25, 27, 28},
                  {28, 28, 29, 29, 29, 30, 31, 31, 32, 32},
                  {32, 33, 33, 35, 36, 37, 38, 39, 39, 42},
                  {42, 43, 43, 43, 44, 45, 49, 52, 52, 54},
                  {54, 55, 55, 56, 58, 60, 60, 61, 62, 62},
                  {64, 67, 68, 68, 70, 71, 72, 72, 73, 75},
                  {76, 77, 78, 81, 81, 84, 84, 85, 86, 86},
                  {86, 87, 90, 90, 91, 95, 96, 97, 97, 98}};

System.out.println("Contains 55?:"+binaryFind(matrix, 55));
System.out.println("Contains 1?:"+binaryFind(matrix, 1));
System.out.println("Contains -1?:"+binaryFind(matrix, -1));
System.out.println("Contains 98?:"+binaryFind(matrix, 98));
System.out.println("Contains 100?:"+binaryFind(matrix, 100));
```

```
Contains 55?:true
Contains 1?:true
Contains -1?:false
Contains 98?:true
Contains 100?:false
```

File to submit: `MatrixSearcher.java`

Question 2 [20 points]

In this project, you are about to implement ICT Dating system. The system can verify valid users and allow all users to send individual or broadcast message to other users. In the system, VIP users have extra feature that can search other user by age. You are given two interface `Chattable` in Chattable.java and `Searchable` in Searchable.java. Your task is to implement TWO classes as following requirements.

Note that: **ICTDatingTester.java** is provided as the main class for testing your program (Do not modify this file). The list of all user in the system are kept in 2d-array variable named *listUser* type integer, where userID, age, and gender are kept as below structure:

listUser

uID ➡	10	1	2	5
Age ➡	15	18	18	20
Gender ➡	1	0	1	0

Task 1 (10 points): Create a java class `User` that use interface `Chattable` in **User.java**:

- `public boolean checkValidUserID(int uID, int [][] listAllUser):` takes integer uID, and 2-D array listAllUser as inputs to check a validity of uID in listAllUser. Then return true if uID is valid in listAllUser, otherwise false.

- `public String msgToUserID(int uID, String message, int [][] listAllUser):` This method will return integer uID and String message if the uID is valid in the listAllUser. For example, for uID=1 and message=" `Test Message`", the method will return **>>** 1|`Test Message`.

- `public String msgToUserIDs(int[] uID, String message, int [][] listAllUser):` This method will return a list of uIDs and String message if the uID is valid in the listAllUser. For example, for uID=1,2,300 and message= "`Test BroadCast`" (where uID 1,2 are valid in the listAllUser), the method will return **>>** 1,2 `Test BroadCast` |300.

Task 2 (10 points): Create a java class `VIPUser` in **VIPUser.java** that have all abilities of class `User` but can perform extra feature of the interface `Searchable`:

- `public int findUserIDByAge(int age, int [][] listAllUser):` this method takes integer age and, 2-D array listAllUser as inputs to search for uID by age of a user. The method will return uID of a user where age of uID matched with input age, otherwise return -1.

Note that: you have to implement **binary search** here. Other method will be marked as zero.

Running the main method of **ICTDatingTester.java** should give the following output:

```
@Test Task1
true
false
1|Test Message
1,2|Test BroadCast|300

@Test Task2
false
0
1,2,3,4|VIP Broadcast|
Find uID of AGE 18 = 20
Find uID of AGE 35 = -1
```

Files to submit: **TWO** files >> *User.java* and *VIPUser.java*